The multi-league sports scheduling problem, or how to schedule thousands of matches

Morteza Davari Dries Goossens Jeroen Beliën
Roel Lambers Frits C.R. Spieksma

February 27, 2020

Abstract

We consider the simultaneous scheduling of multiple sport leagues, with interdependencies arising from teams in different leagues belonging to the same club. Teams from the same club share the same venue with limited capacity. We minimize the total capacity violation in polynomial time when each league has the same, even number of teams. We introduce two generalizations: one where teams from a club have to play according to the same pattern, and one where club capacities differ throughout the season.

This is the peer-reviewed version of https://doi.org/10.1016/j.orl.2020.02.004

1 Introduction

Every sports competition needs a schedule, stating who will play whom, when, and where. Depending on which constraints need to be taken into account, scheduling a single league may already be quite a challenge (see e.g. [Alarcón et al., 2017], [Goossens and Spieksma, 2009], [Recalde et al., 2014]). However, while professional sports usually have only a handful of leagues, in amateur sports or youth competitions, the number of leagues and matches can be very large. For instance, in the Belgian soccer association, each province is responsible for scheduling the matches of its (youth and amateur) teams; a single province may harbor hundreds of clubs, that jointly may have over 5000 teams, distributed over hundreds of leagues, yielding over tens of thousands of matches in one season. In these leagues, clubs typically have several teams (e.g. based on age or skill of the players); however, all teams from the same club share the same infrastructure. This creates a capacity problem at each club: a club has a bound on the number of matches it can host at each point in time (which typically follows from its number of terrains). Observe that these capacity constraints create interdependencies between the leagues, such that it becomes a challenging problem to schedule all leagues while taking these capacities into account.

With respect to scheduling multiple leagues simultaneously, the literature is sparse. Kendall [2008] considers the problem of simultaneously scheduling the matches in four different leagues of the English soccer competition. However, the focus is only on two rounds, played on Boxing day and New Year's day. During these rounds, each team must play one home match and one away match such that the two opponents of each team are different, and that some pairs of teams do not meet at all. In all leagues, the objective is to minimize the total distance traveled by the teams in those two rounds. The solution offered, however, does not generalize to scheduling the entire season. Grabau [2012] describes the scheduling of a recreational softball competition with 74 teams, split over 8 leagues, and competing on 12 fields. The scheduler must adhere to several intertwined scheduling rules, while simultaneously ensuring that the players play their allotment of matches. Burrows and Tuffley [2015] describe a scheduling problem for a competition played in two divisions. The authors try to achieve a maximal number of so-called *common fixtures* between clubs, which occur if their teams in division one and two are scheduled to play each other in the same round. Schönberger [2015] introduces the so-called *championship timetabling problem*, which involves several leagues that are scheduled simultaneously. Two types of inter-league constraints are considered: limited venue capacity as well as player substitution opportunities between several teams of a club. Computational experiments involving a mixed-integer linear program illustrate that even finding a feasible solution for a very small instance with only two leagues of six teams each is a time-consuming task.

In this paper, we study the multi-league scheduling problem as faced by the league organizer. Clearly, when scheduling a single league in professional sports, the precise round in which a particular match takes place can be quite

	Rounds									
	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
h_1	A	H	A	H	A	H	A	H	A	H
h_2	A	H	H	A	H	H	A	A	H	A
h_3	H	A	A	H	A	A	H	H	A	H
h_4	A	H	A	H	H	H	A	H	A	A
h_5	H	A	H	A	A	A	H	A	H	H
h_6	H	A	H	A	H	A	H	A	H	A

Figure 1: A HAPset for a league consisting of 6 teams

important. However, such matters are not relevant when scheduling thousands of matches for hundreds of leagues. In order to cope with this huge number of matches, typically, a league organizer uses the following approach. First, the teams are clustered into leagues of even size. Common practice is to (i) use a geographical clustering, ensuring that teams of the same strength/age category are in a same league, and (ii) to avoid teams of the same club to be present in the same league, see [Toffolo et al., 2019] for a discussion of the problem of grouping teams into leagues. Leagues of even size make sense, as they allow each team to play on each round; and although the total number of teams may not be an exact multiple of the league size, with an even league size the vast majority of the teams will be still able to play each round. Second, the league organizer no longer assigns individual matches to individual rounds. Instead, using a prespecified set of so-called *Home-Away patterns* (in short HAPs, see Section 2 for terminology) that is valid for each league, the league organizer assigns teams to these HAPs. Next, combining this assignment with a compatible *opponent table*, which specifies each team's opponent for each round, the schedule follows.

As an illustration of the latter procedure, consider the HAPset (i.e., a set of HAPs) depicted in Figure 1; it reflects a particular HAPset for a league consisting of 6 teams, where each team plays against each other team twice. Although a priori, the given HAPset may allow different schedules (or none), Figure 2 gives one such schedule compatible with the HAPset from Figure 1. The issue of deciding whether a schedule exists for a given HAPset is a well-researched topic (see [Miyashiro et al., 2003], [Briskorn, 2008], [Horbach, 2010], [Goossens and Spieksma, 2011]); we do not go into details here.

Our contribution focusses exclusively on assigning teams to HAPs. Since such assignment dictates when each team plays home, it specifies for each club how many matches are played at the club's venue in each round. This is important, since the capacity of a club in terms of the number of matches it can host in a round is typically bounded. In fact, a capacity is given for each club; in practice this number follows from the number of available pitches, the set of possible starting times, and the availability of material and referees. Our goal is to find, for each league, an assignment of teams to HAPs minimizing the total capacity violation over the clubs. We refer to the resulting problem as the Multi-league Scheduling Problem (MSP) (see Section 3 for a precise problem

Rounds									
r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
6 vs 1	1 vs 3	2 vs 4	1 vs 2	2 vs 3	1 vs 6	3 vs 1	1 vs 5	2 vs 1	3 vs 2
5 vs 2	2 vs 6	5 vs 1	3 vs 5	4 vs 1	2 vs 5	6 vs 2	4 vs 2	5 vs 3	1 vs 4
3 vs 4	4 vs 5	6 vs 3	4 vs 6	6 vs 5	4 vs 3	5 vs 4	3 vs 6	6 vs 4	5 vs 6

Figure 2: A schedule compatible with the HAPset from Figure 1 where team i has been assigned to h_i, $i = 1, \ldots, 6$

description).

We present a polynomial-time algorithm for the MSP (Section 4). Further, we show that, for a league consisting of at least four teams, the problem becomes difficult when all teams of each club must play according to the same pattern, or when club capacities differ throughout the season (Section 5).

2 Terminology and assumptions

Each team belongs to a club, and each club has a venue. When a team plays at its club's venue, the team plays *home*, otherwise the team plays *away*. A *double round robin tournament* (DRR) is a tournament where each team meets each other team twice. This is a typical format in many sport competitions, such as soccer, basketball, volleyball, hockey; each team meets each other team once home and once away.

When scheduling a tournament, the matches must be allocated to *rounds* in such a way that each team plays at most one match in each round (typically, a round corresponds to a weekend). Since, in our case the number of teams k is even, at least $2(k-1)$ rounds are required to schedule a DRR; if that number is attained, it is called a *compact* DRR.

The sequence of home and away matches according to which a team plays in a tournament, is referred to as a *Home-Away pattern* (in short, HAP). A HAP is represented by a vector consisting of $2(k-1)$ symbols, $k-1$ of which are an 'H', and $k-1$ of which are an 'A'; these obviously refer to the home matches and away matches. A *Home-Away pattern set* (HAPset) corresponds to the set of HAPs, one for each team in the tournament. We say that a HAPset is *feasible* if there exists at least one schedule that is compatible with the HAPset (i.e. for each match i vs. j in round r, team i has an 'H' in its HAP and team j has an 'A').

Two HAPs h and h' are *complementary* if whenever the team assigned to HAP h plays home, the team assigned to HAP h' plays away and vice versa. A *complementary HAPset* is a set that only consists of complementary pairs of HAPs. For example, the HAPset depicted in Figure 1 is a feasible, complementary HAPset with three pairs of complementary HAPs (pair 1: h_1 and h_6; pair 2: h_2 and h_3 pair 3: h_4 and h_5).

In this work, we make a number of assumptions. We assume that each league has the same even number of teams. We also assume that the league organizer uses the same complementary HAPset for each league. This is common practice in competitions where there are few considerations, besides capacity issues. In Section 4, it will become clear that the choice of a particular (complementary) HAPset is irrelevant. Finally, we exclusively deal with compact DRRs for an even number of teams. Consequently, all leagues are played simultaneously, and each team plays either home or away in each round.

3 Problem description

We are given a set T of teams ($n = |T|$), a set L of leagues ($m = |L|$), and a set C of clubs. Also given are two partitions of the set T: one partition $\{\bar{T}_1, ..., \bar{T}_m\}$ of the set T indicates which teams belong to which league; notice that, for each $\ell \in L$, $|\bar{T}_\ell| = k$ with k even since each league consists of the same even number of teams. Another partition of the set T is $\{\hat{T}_1, ..., \hat{T}_{|C|}\}$, which describes which teams belong to which club. Here we have $n_c = |\hat{T}_c|$ for $c \in C$, as clubs can consist of any number of teams. Of course $\sum_{c \in C} n_c = km$. We are also given k HAPs, each of length $2(k-1)$ that jointly form a feasible, complementary HAPset denoted by $\mathcal{H}$. The set of rounds is $\{1, 2, \ldots, 2(k-1)\}$ and is denoted by R. Finally, each club $c \in C$ has a given fixed capacity δ_c, which corresponds to the number of matches it can host in each round.

Capacity violations happen whenever for any club and any round the number of teams of a club that play home exceeds the capacity of the club. The violation of a club in a round is measured by a scalar value that is either zero (if there is no violation) or equal to the number of teams that play home in that round minus the club's capacity (if there is a violation).

The multi-league sports scheduling problem (MSP) is now to find an assignment of teams to HAPs, such that the total capacity violation (i.e. the summation of violations over all clubs and all rounds) is minimized.

Let us introduce binary variables $x_{t,h}$ which equal one if team $t \in T$ is assigned to HAP $h \in \mathcal{H}$ and zero otherwise, and auxiliary variables $z_{c,r}$ that represent the amount of violation of club $c \in C$ in round $r \in R$. An assignment $\mathbf{x}$ is feasible iff the teams in each league are assigned to different HAPs. Given the set of HAPs and the set of rounds, we compute (in a pre-processing step) parameters $U_{h,r}$ which equal one if the team assigned to HAP $h \in \mathcal{H}$ plays home in round $r \in R$, and zero otherwise. The following mixed integer program formulates MSP.

$$
\begin{aligned}
v_{IP} = \min \sum_{c \in C} \sum_{r \in R} z_{c,r} & & (1)\\
\text{s.t.} & & \\
\sum_{t \in \bar{T}_\ell} x_{t,h} = 1 & \quad \forall \ell \in L, h \in \mathcal{H} & (2)\\
\sum_{h \in \mathcal{H}} x_{t,h} = 1 & \quad \forall t \in \bar{T}_\ell, \ell \in L & (3)\\
z_{c,r} \geq \sum_{t \in \hat{T}_c} \sum_{h \in \mathcal{H}} x_{t,h} U_{h,r} - \delta_c & \quad \forall c \in C, r \in R & (4)\\
z_{c,r} \geq 0 & \quad \forall c \in C, r \in R & (5)\\
x_{t,h} \in \{0,1\} & \quad \forall t \in T, h \in \mathcal{H} & (6)
\end{aligned}
$$

This formulation aims to minimize total capacity violation. Constraints (2)-(3) enforce an assignment of teams to HAPs, while Constraints (4)-(5) determine the number of violations of each club in each round. We point out that this mixed integer program can be modified to accommodate situations that are slightly more general than MSP; for instance, situations where each league has its own (given) HAP-set, or where not all leagues play in all rounds can be formulated with minor modifications of (1)-(6).

3.1 The Linear Programming relaxation

When replacing constraints (6) by $x_{t,h} \geq 0$ for each t and h, the LP-relaxation of formulation (1)-(6) arises; we denote the corresponding value by v_{LP}. One might wonder whether the extreme vertices of the polytope corresponding to the LP-relaxation of (1)-(6) are integral. That is not the case, as witnessed by the following example.

Example 1 We have $n = 20$ teams, distributed over $m = 5$ leagues of size $k = 4$, and belonging to six clubs: in this instance, $T = \{t_1, ..., t_{20}\}$, $C = \{c_1, ..., c_6\}$ and $L = \{\ell_1, ..., \ell_5\}$. The partition of teams into clubs, as well as the club's capacities, are given in Figure 3a, and the partition of teams into leagues is given in Figure 3b. The HAPset is as follows:

$$
\mathcal{H} = \left\{ \begin{array}{l} h_1 = \{ \text{ H, A, H, A, H, A } \}, \\ h_2 = \{ \text{ A, H, A, H, A, H } \}, \\ h_3 = \{ \text{ H, A, A, A, H, H } \}, \\ h_4 = \{ \text{ A, H, H, H, A, A } \} \end{array} \right\}.
$$

For this instance, we find as an optimal basic solution to the LP-relaxation

	$\hat{T}_c$	δ_c
c_1:	$\{t_1, t_2, t_3, t_4\}$	2
c_2:	$\{t_5, t_6, t_7\}$	1
c_3:	$\{t_8, t_9, t_{10}\}$	1
c_4:	$\{t_{11}, t_{12}, t_{13}, t_{14}\}$	2
c_5:	$\{t_{15}\}$	1
c_6:	$\{t_{16}, t_{17}, t_{18}, t_{19}, t_{20}\}$	1

(a) Clubs

	T_l
ℓ_1:	$\{t_1, t_6, t_{13}, t_{16}\}$
ℓ_2:	$\{t_2, t_9, t_{12}, t_{17}\}$
ℓ_3:	$\{t_3, t_5, t_{14}, t_{20}\}$
ℓ_4:	$\{t_7, t_8, t_{11}, t_{18}\}$
ℓ_5:	$\{t_4, t_{10}, t_{15}, t_{19}\}$

(b) Leagues

Figure 3: The data associated with Example 1

of (1)-(6):

$$\begin{aligned} x^*_{1,2} =& x^*_{2,1} = x^*_{7,2} = x^*_{8,3} = x^*_{9,4} = x^*_{11,4} = x^*_{12,2} = x^*_{13,1} \\ =& x^*_{14,3} = x^*_{15,1} = x^*_{17,3} = x^*_{18,1} = x^*_{19,2} = x^*_{20,4} = 1, \\ x^*_{3,1} =& x^*_{3,2} = x^*_{4,3} = x^*_{4,4} = x^*_{5,1} = x^*_{5,2} = x^*_{6,3} = x^*_{6,4} = x^*_{10,3} \\ =& x^*_{10,4} = x^*_{16,3} = x^*_{16,4} = 0.5, \end{aligned}$$

while the remaining $x^*_{t,h}$ variables are zero; the values of the $z^*_{c,r}$ variables follow easily. The objective value of this solution to the linear programming relaxation, i.e., $v_{LP} = 15$.

3.2 A combinatorial lower bound

Let us now consider a combinatorial lower bound for v_{IP}. Observe that, in any HAP, there are $k-1$ 'H's. Hence, the total number of home matches of teams belonging to a club $c \in C$ equals $(k-1)n_c$. Total capacity of a club, summed over the rounds, equals $(2k-2)\delta_c$. Clearly, if $(2k-2)\delta_c < (k-1)n_c$, or equivalently, when $\delta_c < \frac{n_c}{2}$, there will be violations for club c. Let us define $C^- = \{c \in C | \ \delta_c < n_c/2\}$. We claim that for each $c \in C^-$ the difference between the number of home matches to be played by teams of club c, and the total capacity of club c is a lower bound for the number of violations of club c, i.e., club $c \in C^-$ will have at least $(k-1)n_c-(2k-2)\delta_c = (2k-2)(\frac{n_c}{2}-\delta_c)$ violations. To capture this number of violations we define the following quantity:

$$Q \equiv 2\,(k-1) \sum_{c \in C^-} \Big(\frac{n_c}{2} - \delta_c\Big). \tag{7}$$

The discussion above implies the following lemma.

Lemma 1 $Q \leq v_{IP}$.

We will show in Section 4 that $Q = v_{LP} = v_{IP}$; in the next theorem, we prove the first of these equalities.

Theorem 1 $Q = v_{LP}$.

Proof: Consider a solution $(\mathbf{x}^*, \mathbf{z}^*)$ that is optimal with respect to the LP-relaxation of (1)-(6). For each club $c \in C$, we have (by summing constraints (4) over the rounds):

$$\sum_{r \in R} z^*_{c,r} \geq \sum_{t \in \hat{T}_c} \sum_{h \in \mathcal{H}} x^*_{t,h} \left(\sum_{r \in R} U_{h,r} \right) - 2(k-1)\delta_c = (k-1)(n_c - 2\delta_c).$$

This implies

$$\sum_{r \in R} z^*_{c,r} \geq \begin{cases} 0 & \text{if } c \in C \setminus C^- \\ 2\,(k-1)\left(\frac{n_c}{2} - \delta_c\right) & \text{if } c \in C^- \end{cases}$$

Thus, the following inequality holds:

$$v_{LP} = \sum_{c \in C} \sum_{r \in R} z^*_{c,r} \geq \sum_{c \in C^-} 2(k-1)(\frac{n_c}{2} - \delta_c) = Q. \tag{8}$$

Now, consider solution $(\hat{\mathbf{x}}, \hat{\mathbf{z}})$ where $\hat{x}_{t,h} = \frac{1}{k}, \forall t \in T, h \in \mathcal{H}$. Due to the fact that the HAPset is feasible, it follows that in each round $\frac{k}{2}$ HAPs have an 'H', while the remaining $\frac{k}{2}$ HAPS have an 'A'. Thus, for each $t \in T$, we have that $\sum_{h \in \mathcal{H}} \hat{x}_{t,h} = \frac{1}{k} \cdot \frac{k}{2} = \frac{1}{2}$, and hence the (fractional) number of home matches of a club $c \in C$ in each round equals $\frac{n_c}{2}$, leading to a violation in a round equaling: $\max\left\{\frac{1}{2}n_c - \delta_c, 0\right\}$. Thus, the objective value of this solution is exactly Q, which implies $v_{LP} \leq Q$. Together with (8), the result follows. □

In light of Theorem 1, one may wonder whether it is possible to round an optimal, fractional LP-solution into an optimal integral solution. That, however, does not seem to be straightforward.

Indeed, consider the LP-solution discussed in Example 1, a solution in which there are no violations for club c_1. The non-zero variables associated to teams of club c_1 are $x^*_{1,2} = x^*_{2,1} = 1$ and $x^*_{3,1} = x^*_{3,2} = x^*_{4,3} = x^*_{4,4} = 0.5$. A straightforward rounding of this solution would imply that teams t_1 and t_2 are assigned to complementary HAPs h_1 and h_2, and therefore teams t_3 and t_4 should also be assigned to complementary HAPs (otherwise, club c_1 will have violations in some rounds). But that cannot be achieved by any straightforward procedure that rounds the current fractional solution.

In the next section we provide a polynomial-time algorithm that finds an optimal solution to MSP.

4 A polynomial-time, exact algorithm for MSP

In this section, we exhibit Algorithm 1 that outputs an optimal solution to MSP in polynomial time. Interestingly, the values of the capacities δ_c do not impact the solution; in other words, the solution found by Algorithm 1 is optimal for *any* capacities δ_c. Informally, this solution is one where the home matches of

Algorithm 1

Input: An instance of MSP.

1: Create a new instance of MSP as follows. Partition, arbitrarily, each club c into $\lfloor n_c/2 \rfloor$ arbitrary pseudo clubs of size two, and add the remaining team (if there is one) to a new club c'.

2: Partition club c' into $n_{c'}/2$ arbitrary pseudo clubs of size two. Set the capacity of all pseudo clubs in the new instance to one. Notice that in the new instance the assignment of teams to leagues remains unchanged.

3: Construct the following graph G based on the new instance: there is a vertex for each league $\ell \in L$ and there is an edge for each pseudo club that links the two vertices/leagues where the two teams of that pseudo club play. Note that G will be a multi-graph with m vertices and $km/2$ edges.

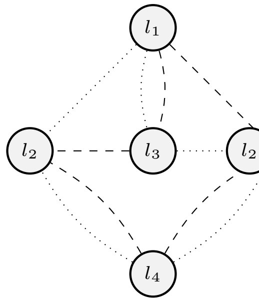
4: Consider a 2-factorization of G. Associate each 2-factor with a pair of complementary HAPs. This leads to a feasible assignment $\mathbf{x}$: in each 2-factor, each edge is associated with two teams from a pseudo club, and each vertex is associated with two teams in a league that follow the associated pair of complementary HAPs.

Output: An assignment of teams to HAPs: $\mathbf{x}$.

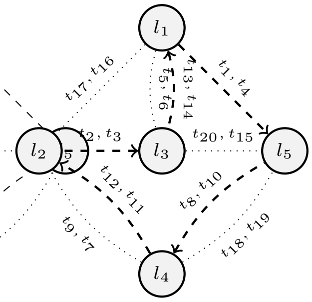
teams from the same club are as balanced over the rounds as possible. Before proving correctness of Algorithm 1, we first illustrate how Algorithm 1 works on the instance given in Example 1.

Following steps 1 and 2 of Algorithm 1, we first create a new instance. The clubs c_1 and c_4 consist of an even numbers of teams and thus we can split them into four clubs (so-called pseudo clubs) of two teams as follows: $\hat{T}_{c_{1a}} = \{t_1, t_4\}$, $\hat{T}_{c_{1b}} = \{t_2, t_3\}$, $\hat{T}_{c_{4a}} = \{t_{11}, t_{12}\}$ and $\hat{T}_{c_{4b}} = \{t_{13}, t_{14}\}$. The remaining clubs consist of an odd numbers of teams. For instance club c_6 consists of five teams, therefore we split it into two clubs of size two and add the remaining team to the new club c'. Hence, $\hat{T}_{c_{6a}} = \{t_{16}, t_{17}\}$ and $\hat{T}_{c_{6b}} = \{t_{18}, t_{19}\}$ and team t_{20} is added to club c'. We repeat the process for the other clubs with odd numbers of teams. As a result, we have $\hat{T}_{c_{2a}} = \{t_5, t_6\}$, $\hat{T}_{c_{3a}} = \{t_8, t_{10}\}$ and $\hat{T}_{c'} = \{t_7, t_9, t_{15}, t_{20}\}$. Finally, we split club c' as follows: $\hat{T}_{c'_a} = \{t_7, t_9\}$ and $\hat{T}_{c'_b} = \{t_{15}, t_{20}\}$.

Then, in steps 3 and 4 of Algorithm 1, we construct a graph G and identify a 2-factorization of G (see Figure 4a; recall that a 2-factor is a collection of cycles spanning all vertices of G, and a 2-factorization is a partitioning of the edges of G into 2-factors). Now, pick one of the 2-factors and associate it with pair (h_1, h_2) and the other 2-factor with pair (h_3, h_4). To assign teams to HAPs we start with one arbitrary team that is visited in the first 2-factor, for instance team t_1, and assign it to h_1. We traverse the 2-factor in an arbitrary direction (starting from the edge containing team t_1) and enforce the two teams associated to each edge to HAPs h_1 and h_2 (see Figure 4b). Thus, if $t_1 \to h_1$ (t_1 is assigned to h_1), $t_4 \to h_2$, $t_{10} \to h_1$, $t_8 \to h_2$, $t_{11} \to h_1$, $t_{12} \to h_2$, $t_2 \to h_1$, $t_3 \to h_2$, $t_{14} \to h_1$ and $t_{13} \to h_2$. Similarly we assign the teams in the other 2-factor to HAPs h_3 and h_4 ($t_7 \to h_3$, $t_9 \to h_4$, ...).



(a) A 2-factorization of G



(b) Labeled and directed G

Figure 4: The graph associated with Example 1

The capacities of clubs c_1, c_4 and c_5 are never violated. Club c_2 has one-unit violations at rounds 1, 5 and 6; club c_3 has one-unit violations at rounds 2, 3 and 4; club c_6 has one-unit violations at rounds 2, 3 and 4 and two-unit violations at rounds 1, 5 and 6. The total violation for this solution is 15.

Theorem 2 *Algorithm 1 solves* MSP *in* $\mathrm{O}(nm)$*-time.*

Proof: We first comment on the different steps in Algorithm 1. Clearly, since the league size k is even, the construction in Steps 1 and 2 implies that each pseudo club contains exactly two teams. Further, the construction in Step 3 implies that G is k-regular, and thus 2-factorable (since k is even). Notice that the case where two teams of the same club play in the same league amounts to a loop in G, and will result in these two teams receiving complementary HAPs. We refer to Lovász and Plummer [1986, Theorem 6.2.4] for details regarding finding such a 2-factorization.

We now show correctness of Algorithm 1. Consider a solution outputted by Algorithm 1. Each pair of teams that make up a pseudo club use complementary patterns, and hence, they jointly play one home match in each round. Thus, if $\delta_c \geq \frac{n_c}{2}$, i.e., if club $c \in C \setminus C^-$, there are no violations for club c. In addition, if $\delta_c < \frac{n_c}{2}$, i.e., if $c \in C^-$ then the number of violations of club c equals:

$$2\,(k-1)\,(n_c/2-\delta_c)\,.$$

Using (7), it follows that the value of the solution found by Algorithm 1 equals Q, and is thereby necessarily optimal. Note that this implies the second equality of Theorem 1.

To establish the complexity of Algorithm 1, observe that in the first step, a new instance is generated where each club consists of exactly two teams. This is done in $\mathrm{O}(n)$-time. In the second and third step, a graph G is constructed which is done in $\mathrm{O}(n)$-time and then a 2-factorization of G is computed which is done in $\mathrm{O}(nm)$-time (see the algorithm provided by Lovász and Plummer

[1986, Theorem 6.2.4]). Finally, the 2-factorization is mapped to a solution for the original instance, which is done in $O(n)$-time. Therefore, the algorithm runs in $O(nm)$-time. □

Interestingly, from the proof of Theorem 2, we observe that the given HAPset $\mathcal{H}$ (as long as it is complementary) has no impact, neither on the optimal solution nor on the minimum violation. Further, the proofs of Theorems 1 and 2 imply the following corollary.

Corollary 1 $Q = v_{LP} = v_{IP}$.

5 Two generalizations of MSP

In this section, we investigate two generalizations of MSP. In Section 5.1, we consider an extension of MSP where all teams from the same club must play according to the same HAP; we refer to this generalization as MSPidHAP. Next, in Section 5.2, we deal with an extension of MSP in which capacities are not necessarily constant over the rounds, which we call MSPwVC. We motivate both generalizations, and we show that both problems are NP-hard for $k \geq 4$, and give polynomial-time algorithms for the case $k = 2$, when, in case of MSPwVC, each club consists of two teams. Observe that a league size of $k = 2$ may occur in knock-out tournaments, or play-offs, where two matches decide upon the winner of a pair of teams.

5.1 MSP with identical HAPs (MSPidHAP)

In a setting where the capacity of clubs is not an issue, clubs may want that *all* their teams play home in the same round. There can be various reasons for this wish: for instance to create a lively atmosphere at the club's venues, or to minimize the number of times a venue is used, or, when clubs have two or more teams in one particular category (for instance a club has two amateur teams in the under 21-years-old age category), teams following the same HAP allow these teams to exchange players whenever they play home.

The input defining an instance of MSPidHAP consists of the set of teams, its two partitions (one into leagues, and one into clubs), and a feasible, complementary HAPset. The question is: does there exist a *feasible assignment*, i.e., does there exist an assignment of teams to HAPs such that (i) all teams from a club receive the same HAP, and (ii) all teams from a league receive a different HAP? Of course, in an instance of MSPidHAP, it should not happen that two teams from a same club are in the same league, since this would clearly lead to a no-instance.

It is not difficult to see that, in case $k = 2$, this question can be answered efficiently as follows: build a simple undirected graph $G = (V, E)$ with a vertex for each club ($V = C$), and connect two vertices iff the corresponding clubs have a team in a same league ($E = L$). The existence of a feasible 2-coloring of the vertices of G decides whether or not the instance of MSPidHAP with $k = 2$

is a yes-instance or not. It is a fact that all teams of clubs corresponding to nodes colored with one color play according to HAP HA, and all teams of clubs corresponding to nodes colored with the other color play according to HAP AH. We record this observation formally.

Observation 1 *For $k = 2$, MSPidHAP is solvable in polynomial time.*

It is possible to extend Observation 1 to a situation where a set of pairs of teams that need the same HAP is given. However, when $k \geq 4$, MSPidHAP becomes more difficult.

Theorem 3 *MSPidHAP is NP-hard for each $k \geq 4$.*

Proof: We reduce MSPidHAP to edge coloring a 4-regular graph. In this reduction, we do not explicitly construct a feasible, complementary HAPset. In fact, we assume that some HAPset is specified; the proof works for any given HAPset.

Consider now the following question: given a simple 4-regular graph $G = (V, E)$, does there exist a coloring of the edges using 4 colors such that no two adjacent edges receive the same color? This problem is known to be strongly NP-complete [Holyer, 1981, Leven and Galil, 1983].

Given a simple 4-regular graph $G = (V, E)$, we construct an instance of MSPidHAP as follows. There is a league $\ell \in L$ for each vertex in V, i.e., $L = V$. There is a club $c \in C$ for each edge $e = (v, v') \in E$, i.e., $C = E$; each club consists of two teams ($n_c = 2$), one playing in the league corresponding to node v, one playing in the league corresponding to node v'. Thus, there are $n = 2|E|$ teams. We claim that the existence of a 4-coloring of G corresponds to a feasible assignment of teams to HAPs and vice versa.

Suppose that a 4-coloring exists. Let each color correspond to a HAP. By assigning the two teams of a club to the HAP that corresponds to the color of the edge corresponding to those two teams, it becomes clear that the feasibility of the coloring implies that the four teams in each league have received a pairwise different HAP, and hence a feasible assignment exists.

Suppose a feasible assignment exists. Then all teams that play according to HAP i receive color i, $i = 1, \ldots, 4$; this results in a 4-coloring of G. $\square$

5.2 MSP with variable capacities (MSPwVC)

Another generalization of MSP is the problem where clubs' capacities differ throughout the season. We refer to this problem as *MSP with variable capacities* (in short MSPwVC). In this generalization, instead of having a constant capacity δ_c for a club, we are given capacities $\delta_{c,r}$ that represents the number of matches that can be hosted by club c in round r. The resulting problem can be formulated as an integer program by replacing Constraints (4) by:

$$z_{c,r} \geq \sum_{t \in \hat{T}_c} \sum_{h \in \mathcal{H}} x_{t,h} U_{h,r} - \delta_{c,r} \qquad \forall c \in C, r \in R. \tag{9}$$

$\delta_{i,1}, \delta_{i,2}$	0	1	2
0	$\{(2,0),(1,1),(0,2)\}$	$\{(1,1),(0,2)\}$	$\{(0,2)\}$
1	$\{(2,0),(1,1)\}$	$\{(1,1)\}$	$\{(1,1),(0,2)\}$
2	$\{(2,0)\}$	$\{(2,0),(1,1)\}$	$\{(2,0),(1,1),(0,2)\}$

Figure 5: Ideal occupations $(o_{i,1}, o_{i,2})$ of club i when given capacities $\delta_{i,1}, \delta_{i,2}$

The resulting formulation of MSPwVC becomes:

$$\left\{ \min \sum_{c \in C} \sum_{r \in R} z_{c,r} \;\middle|\; (2), (3), (5)-(6), (9) \right\}.$$

In Section 5.2.1, we provide, for the case where $k = n_c = 2$, a polynomial-time algorithm based on finding a min-cost circulation, and in Section 5.2.2 we show that the problem becomes NP-hard for $k \geq 4$.

5.2.1 MSP with variable capacities: the case $k = 2$

Consider an instance of MSPwVC consisting of clubs C, leagues L, teams T, capacities $(\delta_{c,1}, \delta_{c,2})$, that features $k = n_c = 2$ for all $c \in C$. Note that for this specific setting $m = |L| = |C|$. Let us first argue that we can restrict our attention to instances that are "connected", as explained hereunder. Indeed, we can represent such an instance by building a bipartite graph $H = (V_1 \cup V_2, E)$, where $V_1 = C$, $V_2 = L$ and $E = T$; thus, an edge $(v_1, v_2) \in E$ represents that a team from the club represented by $v_1 \in V_1$ plays in the league represented by $v_2 \in V_2$. As $k = n_c = 2$ for all $c \in C$, the degree of each node in H equals 2, and hence the graph H consists of a collection of disjoint cycles. Clearly, we can restrict our attention to instances where H is a single cycle; we assume, without loss of generality, that by rearranging indices we have a set of clubs $C = \{c_1, c_2, \ldots, c_m\}$ such that each club $c_i, i = 1, \ldots, m-1$ has a team in league ℓ_i and a team in league ℓ_{i+1} and club c_m has a team in league ℓ_m and a team in league ℓ_1.

As the league size is $k = 2$, there are only two different HAPs a team can have, either HA or AH. As every club has only 2 teams, the capacity $\delta_{i,r}$ of club $c_i, 1 \leq i \leq m$, in round $r = 1, 2$ can be seen as either $0, 1$ or ≥ 2. In fact, capacities whose value exceeds 2 can be set to 2 without any consequences; we thus assume $\delta_{i,r} \in \{0, 1, 2\}$ for all $i = 1, \ldots, m$, $r = 1, 2$. It follows that for a particular club c_i there are nine possibilities for $(\delta_{i,1}, \delta_{i,2})$.

A solution to MSPwVC with $k = 2$ can be described as an *occupation* $(o_{i,1}, o_{i,2})$ specifying how many teams of club c_i play home in rounds 1 and 2 respectively; clearly $(o_{i,1}, o_{i,2}) \in \{(2,0), (1,1), (0,2)\}$. We say that an occupation is *ideal* for club c_i when it results in a minimum number of violations over the two rounds given its capacities $\delta_i = (\delta_{i,1}, \delta_{i,2})$. Figure 5 gives, for each of the nine possibilities for $(\delta_{i,1}, \delta_{i,2})$ the set of ideal occupations.

From Figure 5, we see that the occupation $(o_{i,1}, o_{i,2}) = (1,1)$ is ideal for all capacities except when $(\delta_{i,1}, \delta_{i,2}) = (2,0)$ or $(0,2)$, $1 \leq i \leq m$. This observation forms the basis of our approach which, informally said, will use the occupation $(o_{i,1}, o_{i,2}) = (1, 1)$ for each club c_i as a baseline solution, and next will find a maximum number of saved violations by modifying the occupation of appropriately chosen clubs to $(2, 0)$ or $(0, 2)$.

We now describe the construction of a directed graph $G = (V, A)$ that is instrumental in our procedure to solve the problem. The vertex set consists of $V = L \cup \{v_0\}$, where vertex v_i corresponds to league $\ell_i \in L$, $(i = 1, \dots, m)$. The arc set $A = A_1 \cup A_2 \cup A_3$ is defined as follows:

$$A_1 = \{(v_i \to v_{i+1}) : i = 1, \dots, m-1\} \cup \{(v_m \to v_1)\},$$
$$A_2 = \{(v_0 \to v_i) : i = 1, \dots, m\} \text{ and } A_3 = \{(v_i \to v_0) : i = 1, \dots, m\}.$$

To each arc $a \in A$, we associate a capacity $\text{cap}(a)$, and a cost-coefficient $\text{cost}(a)$. We set $\text{cap}(a) = 1$ for each $a \in A$. The costs are defined as follows:

- for each $a \in A_1$, $\text{cost}(a) = 0$,
- for each $a_{0,i} = (v_0 \to v_i) \in A_2$ $(1 \le i \le m)$,

$$\text{cost}(v_0 \to v_i) = \begin{cases} -1 & \text{if } \delta_i = (2, 0), \\ 0 & \text{if } \delta_i \in \{(0, 0), (1, 0), (2, 1), (2, 2)\}, \\ 1 & \text{if } \delta_i \in \{(0, 1), (0, 2), (1, 1), (1, 2)\}. \end{cases}$$

- for each $a_{i,0} = (v_i \to v_0) \in A_3$ $(1 \le i \le m)$,

$$\text{cost}(v_i \to v_0) = \begin{cases} -1 & \text{if } \delta_i = (0, 2), \\ 0 & \text{if } \delta_i \in \{(0, 0), (0, 1), (1, 2), (2, 2)\}, \\ 1 & \text{if } \delta_i \in \{(1, 0), (1, 1), (2, 0), (2, 1)\}. \end{cases}$$

We claim that this particular definition of the cost-coefficients for arcs in A_2 (respectively, A_3) captures the number of violations saved when instead of using occupation $(1, 1)$ occupation $(2, 0)$ (respectively, $(0, 2)$) is used for club c_i with capacity $\delta_i = (\delta_{i,1}, \delta_{i,2})$ - this claim can be verified using the entries given in Figure 5. Indeed, as an example, if the capacity of some club c_i equals $(2, 0)$, then the number of violations saved when using occupation $(2, 0)$ instead of occupation $(1, 1)$ equals 1; this is reflected in the -1 value of $\text{cost}(v_0 \to v_i)$ when $\delta_i = (2, 0)$. Figure 6 depicts the above-described graph G.

We now state Algorithm 2 that computes a minimum cost circulation in graph G. Recall that a circulation is a flow such that, for each node, the amount of flow entering the node equals the amount of flow leaving the node. Obtaining a minimum cost circulation can be done in polynomial time, see [Ahuja et al., 1993].

Theorem 4 *Algorithm 2 solves* MLSwVC *in polynomial time when $k = 2$ and $n_c = 2$ for $c \in C$.*

Proof: The value of a solution to an instance of MLSwVC with $k = 2$ and $n_c = 2$ for $c \in C$, is nothing else but the total number of violations induced by

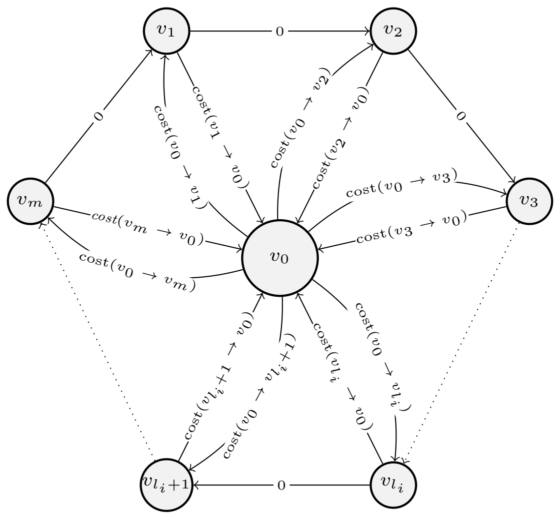


Figure 6: Directed graph G in Section 5.2.1.

Algorithm 2

Input: Clubs C, Teams T, Leagues L, capacities $(\delta_{i,1}, \delta_{i,2})$

1: Build graph G as described above.
2: Solve a min-cost circulation problem on G, getting flow $y(a) \in \{0, 1\}$ for each arc $a \in A$.
3: Set $x_i := 1$ for $i = 1, \dots, m$.
4: For each arc $(v_0 \to v_i) = a \in A_2$ for which $y(a) = 1$: (i) $x_i := 0, j := i$ (ii) WHILE $y(v_j \to v_{j+1}) = 1$ DO $x_j := 0, j := j + 1$.

Output: $(x_1, \dots, x_m)$, where $x_i := 0$ (1) indicates that in league ℓ_i the team from club c_i (c_{i-1}) first plays at home.

the occupations of the clubs. Consider a solution where each club has occupation (1,1) - we will refer to this solution as the *baseline* solution, and we denote its value by B. Further, let the value of a minimum cost circulation in G (found in Step 2 of Algorithm 2) be denoted by q (notice that $q \leq 0$ because there is always a circulation with no flow, i.e., $y(a) = 0, a \in A$). We prove the theorem by showing an equivalence between a minimum cost circulation in G with value q, and the existence of a solution with value $B + q$.

$\Rightarrow$ Consider an optimum solution to the circulation problem in G. Since, for each $i = 1, \dots, m$:

$$\text{cost}(v_0 \to v_i) + \text{cost}(v_i \to v_0) \geq 0,$$

it follows that there exists an optimum solution that does not have a unit flow using the two arcs $(v_0 \to v_i)$ and $(v_i \to v_0)$. Hence, an optimum circulation consists of cycles in G, each cycle carrying one unit of flow, such that each node of G, except v_0, occurs in at most one cycle; such a cycle can be expressed as follows: $v_0, v_i, v_{i+1}, \dots, v_j, v_0$. The cost of an individual cycle depends solely on the costs of the two arcs (v_0, v_i) and (v_j, v_0). Notice that these costs represent, by definition, the savings in the number of violations when the occupation of club c_i (c_j) becomes (2,0) ((0,2)) instead of (1,1). Thus, a circulation in G with cost q leads to a solution of the problem with cost $B + q$.

$\Leftarrow$ Now we show that any solution of our problem corresponds to a circulation in the graph G. As described before, a solution can be seen as the set of occupations for the clubs. Let us associate the occupation of club c_i to node v_i in G. We claim that a solution is feasible iff occurrences of occupations (2,0) and (0,2) alternate along the cycle defined by the arcs in A_1. Given this fact, we can associate a circulation to each solution as follows.

Let x_i denote the schedule of the league ℓ_i, where $x_i = 0$ indicates that league ℓ_i has a schedule in which the team from club c_i first plays at home, and $x_i = 1$ if the league has a schedule in which the team from club c_{i-1} will first play at home. Notice that the occupation of a club c_i can be expressed as $(1 - x_i + x_{i-1}, 1 + x_i - x_{i-1})$.

Given any solution $x = (x_1, \dots, x_n)$, we create a flow y on the edges of the graph G in the following way. We define index sets $I_1 = \{i : x_i = 0, x_{i-1} = 1\}$, $I_2 = \{i : x_i = 0\}$ and $I_3 = \{i : x_i = 1, x_{i-1} = 0\}$ - with $x_0 = x_n$. For every $i \in I_1$, set $y(v_0 \to v_i) = 1$. For $i \in I_2$, set $y(v_i \to v_{i+1}) = 1$. For $i \in I_3$, set $y(v_i \to v_0) = 1$. This results in a flow through the graph with cost $\sum_{i \in I_1} cost(v_0 \to v_i) + \sum_{i \in I_3} cost(v_i \to v_0)$.

All clubs $i \in I_1$ have occupation $(2, 0)$, all clubs in I_3 have occupation $(0, 2)$, while all other clubs have occupation $(1, 1)$. By construction of the graph, the cost of the circulation corresponds exactly with the difference in capacity violation of the solution x compared to a schedule in which all clubs have occupation $(1, 1)$. Therefore, minimizing the cost of the circulation minimizes the number of violations. Hence, Algorithm 2 is an exact algorithm. □

5.2.2 MSP with variable capacities: the case $k \geq 4$

We now show that MSPwVC becomes NP-hard when $k \geq 4$.

Theorem 5 MSPwVC *is NP-hard for each* $k \geq 4$.

Proof: For our reduction we use a problem known as the *restricted timetabling problem* (in short, RTT), proven to be NP-complete in Even et al. [1976]. We first describe the RTT using the terminology from [Even et al., 1976]. We are given a set of exactly three time slots (hours) $\Pi = \{\pi_1, \pi_2, \pi_3\}$, a set of teachers $\mathcal{T}$ and a set of classes V (a class refers to a group of students). Classes are always available, whereas teachers have a given availability, i.e., for each teacher $\tau \in \mathcal{T}$, there is a set of available time slots $\Pi_\tau \subseteq \Pi$. We are also given a set S of courses, each of which must be taught by a specific teacher τ to a specific class ν during anyone of the three time slots. We denote courses by pairs (τ, ν). At most three courses are taught to each class and every teacher is either a *tight 2-teacher* or a *tight 3-teacher*. A teacher is a *tight α-teacher* if he/she teaches exactly α courses and is available for exactly α time slots, $\alpha \in \{2, 3\}$. Let us denote the number of courses taught to a class ν by $\rho(\nu)$, $\nu \in V$. The question is whether there exists an assignment of time slots to each course (τ, ν) such that teachers' availabilities are satisfied and there are no overlaps (i.e., the courses taught by the same teacher are assigned to different time slots and the courses corresponding to each class are also assigned to different time slots).

Given an instance of RTT, we construct an instance of MSPwVC with clubs C, leagues L, teams T and capacities $\delta_{c,r}$ as follows. Each class $\nu \in V$ is associated with a league $\ell \in L$ and thus our instance has $m = |V|$ leagues. Our instance has $\sum_{\nu \in V}(k - \rho(\nu)) + |\mathcal{T}|$ clubs: we associate a club of α teams to each tight α-teacher $\tau \in \mathcal{T}$ (the resulting set of clubs is denoted by C_1); the remaining $\sum_{\nu \in V}(k-\rho(\nu))$ clubs each have exactly one team (these clubs belong to subsets C_2 and C_3 such that $|C_2| = m(k-3)$ and $|C_3| = \sum_{\nu \in V}(3 - \rho(\nu))$; note that $C = C_1 \cup C_2 \cup C_3$). Our instance thus has $\sum_{\nu \in V}(k-\rho(\nu)) + |S|$ teams. Each course $(\tau, \nu) \in S$ represents a team $t \in T$ that belongs to a club in C_1 which is associated with teacher $\tau \in \mathcal{T}$ and plays in the league corresponding to class $\nu \in V$. We distribute the teams of clubs in C_2 by placing $k-3$ teams of clubs in C_2 in each of the leagues. The remaining $\sum_{\nu \in V}(3 - \rho(\nu))$ teams are members of clubs $c \in C_3$; we arbitrarily add these teams to leagues such that all leagues consist of k teams.

Consider a given complementary HAPset $\mathcal{H} = \{h_1, ..., h_k\}$ with complementary pairs $(h_{2j-1}, h_{2j}), j = 1, ..., \frac{k}{2}$. We determine the capacity of clubs $c \in C_1$ as follows: first, we associate the HAP h_κ to time slot π_κ for $\kappa = 1, 2, 3$. Then for each club $c \in C_1$, we identify the set of HAPs which correspond to the time slots during which the teacher (that gave rise to club $c \in C_1$) is available. Recall that each teacher is available either in time slots $\{\pi_1, \pi_2, \pi_3\}$, or $\{\pi_1, \pi_2\}$, or $\{\pi_1, \pi_3\}$, or $\{\pi_2, \pi_3\}$. The capacity of a club $c \in C_1$ is determined by the available time slots. We have, for each $c \in C_1, r \in R$:

$$\delta_{c,r} = \sum_{h \in \mathcal{H}_c} U_{h,r}, \tag{10}$$

where $\mathcal{H}_c$ equals either $\{h_1, h_2, h_3\}$ or $\{h_1, h_2\}$, or $\{h_1, h_3\}$, or $\{h_2, h_3\}$, depending on the availabilities of the teacher giving rise to club $c \in C_1$.

We determine the capacity of a club $c \in C_2$ as follows. We partition C_2 into $k-3$ subsets $C_2^1, ..., C_2^{k-3}$ each containing m clubs such that the teams belonging to the clubs of subset $C_2^i, i = 1, ..., k-3$ all play in different leagues. Next, we set for each club $c \in C_2^i$, $i = 1, \ldots, k-3$, and each round $r \in R$:

$$\delta_{c,r} = U_{h_{i+3},r}.$$

Finally, for each club $c \in C_3$, we set $\delta_{c,r} = 1$ for each round $r \in R$. This completes the description of an instance of MSPwVC.

We now show that a solution to MSPwVC without any capacity violations corresponds to a yes-instance of RTT and vice versa. Suppose that the instance of MSPwVC admits a solution without any capacity violation. In such a solution it must be the case that each team from a club in C_2 has been assigned the one

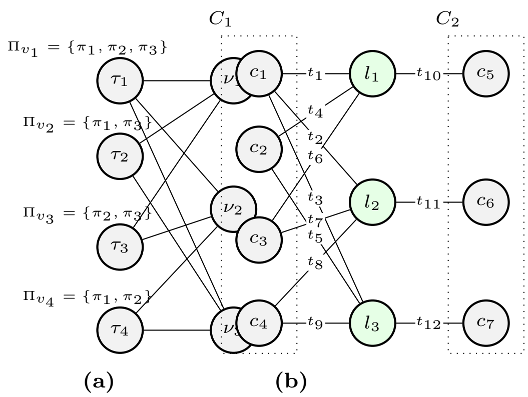


Figure 7: Graph representations for the proof of Theorem 5; (a) A graph representation for the instance of RTT. (b) a graph representation of the instance of MSPwVC.

HAP in the HAPset that yields no capacity violation for this club; in other words, each team from club $c \in C_2^i$ is assigned to HAP h_{i+3} for $i = 1, \ldots, k-3$. Consider now the teams from a club $c \in C_1$. This club has a capacity given by (10) which must be fully utilized in order to have no capacity violations. Hence, the only set of patterns that satisfy this requirement are those patterns in $\{h_1, h_2, h_3\}$ that correspond with club $c \in C_1$, and we assign the teams accordingly. Teams from clubs in C_3 receive any remaining pattern. Based on this assignment of teams to HAPs, we can assign time slots to courses in RTT. The resulting assignment is feasible since (1) each team in a league is assigned to a different HAP, thus the courses taught to each class are assigned to different time slots, (2) the teams from a single club are assigned to different HAPs, thus the courses taught by the associated teacher are assigned to different time slots.

If the instance of RTT is a yes-instance, we simply copy the existing assignment of courses to time slots to the instance of MSPwVC, where the assignment of teams of clubs in C_1 to the given HAPs h_1, h_2, h_3 follows directly from the solution to the RTT instance. Further, we give each team of a club in C_2 its corresponding pattern, and each team in C_3 any remaining pattern. This gives no capacity violations in the instance of MSPwVC. $\square$

As an illustration of the reduction in Theorem 5, consider the following instance of RTT: there are four teachers $(\tau_1, \tau_2, \tau_3, \tau_4)$, three classes (ν_1, ν_2, ν_3) and three time slots (π_1, π_2, π_3). Teacher τ_1 teaches different courses to all classes and is available on all time slots. Teacher τ_2 teaches only to classes ν_1 and ν_3 and is available on time slots π_1 and π_3. Teacher τ_3 teaches only to classes ν_1 and ν_2 and is available on time slots π_2 and π_3. Finally, teacher τ_4 teaches only to classes ν_2 and ν_3 and is available on time slots π_1 and π_2. Figure 7a shows a graph representation of this instance.

Assuming $k = 4$, we construct an instance of MSPwVC with 3 leagues, 7 clubs and 12 teams: $L = \{\ell_1, \ell_2, \ell_3\}$, $C = \{c_1, ..., c_7\}$ and $T = \{t_1, ..., t_{12}\}$ where

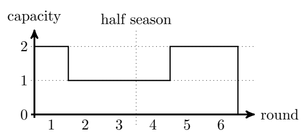
		$\hat{T}_c$	$\boldsymbol{\delta}_c$
C_1	c_1:	$\{t_1, t_2, t_3\}$	Figure 9a
	c_2:	$\{t_4, t_5\}$	Figure 9b
	c_3:	$\{t_6, t_7\}$	Figure 9c
	c_4:	$\{t_8, t_9\}$	Figure 9d
C_2	c_5:	$\{t_{10}\}$	Figure 9e
	c_6:	$\{t_{11}\}$	Figure 9e
	c_7:	$\{t_{12}\}$	Figure 9e

(a) Club in I

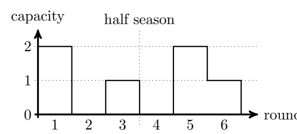
	T_ℓ
ℓ_1:	$\{t_1, t_4, t_6, t_{10}\}$
ℓ_2:	$\{t_2, t_7, t_8, t_{11}\}$
ℓ_3:	$\{t_3, t_5, t_9, t_{12}\}$

(b) Leagues in I

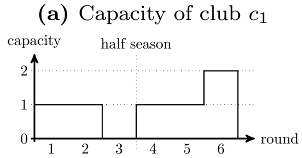
Figure 8: The instance I associated with the example in Theorem 5



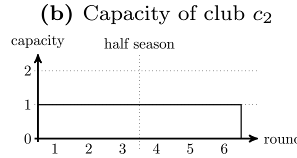
(a) Capacity of club c_1



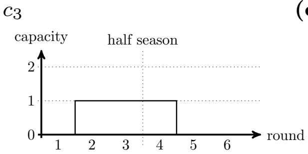
(b) Capacity of club c_2



(c) Capacity of club c_3



(d) Capacity of club c_4



(e) Capacity of clubs c_5, c_6 and c_7

Figure 9: Capacity of the associated clubs

$C_1 = \{c_1, c_2, c_3, c_4\}$ and $C_2 = \{c_5, c_6, c_7\}$. The clubs and leagues are thus given in Figure 8a and Figure 8b. Also, Figure 7b provide a graph representation of the clubs, leagues and teams. The capacity profiles of clubs are given in Figure 9 and are based on the HAPset given in Example 1. As an example, the capacity profile of club c_2 (that is associated with teacher τ_2 for which $\Pi_{\tau_2} = \{\pi_1, \pi_3\}$) is $\boldsymbol{\delta}_c = (U_{h_1,1} + U_{h_2,1}, ..., U_{h_1,6} + U_{h_2,6}) = (1+1, ..., 0+1) = (2, 0, 1, 0, 2, 1)$ (see Figure 9b).

There is a solution with objective value of zero for this instance which is obtained by assigning $h_1 \to t_2, t_4, t_9$, $h_2 \to t_3, t_6, t_8$, $h_3 \to t_1, t_5, t_7$, and $h_4 \to t_{10}, t_{11}, t_{12}$. Hence, the given instance of RTT is a yes-instance.

Acknowledgements: The research of Frits C.R. Spieksma was partly funded by the NWO Gravitation Project NETWORKS, Grant Number 024.002.003.

References

R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows*. Prentice Hall, 1st edition, 1993.

F. Alarcón, G. Durán, M. Guajardo, J. Miranda, H. Muñoz, L. Ramírez, M. Ramírez, D. Saure, M. Siebert, S. Souyris, A. Weintraub, R. Wolf-Yadlin, and G. Zamorano. Operations research transforms the scheduling of Chilean soccer leagues and South American World Cup qualifiers. *Interfaces*, 47:52 – 69, 2017.

D. Briskorn. Feasibility of home–away pattern sets for round robin tournaments. *Operations Research Letters*, 36(3):283–284, 2008.

W. Burrows and C.P. Tuffley. Maximising common fixtures in a round robin tournament with two divisions. *Australasian Journal of Combinatorics*, 63: 153 – 169, 2015.

S. Even, A. ltai, and S. Shamir. On the complexity of timetable and multi-commodity flow problems. *SIAM Journal on Computing*, 5(4):691 – 703, 1976.

D. Goossens and F.C.R. Spieksma. Scheduling the Belgian football league. *Interfaces*, 39(2):109 – 118, 2009.

D. Goossens and F.C.R. Spieksma. Breaks, cuts, and patterns. *Operations Research Letters*, 39:428 – 432, 2011.

K. Grabau. Softball scheduling as easy as 1-2-3 (strikes you're out). *Interfaces*, 42:310 – 319, 2012.

I. Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10:718 – 720, 1981.

A. Horbach. A combinatorial property of the maximum round robin tournament problem. *Operations Research Letters*, 38(2):121–122, 2010.

G. Kendall. Scheduling English football fixtures over holiday periods. *Journal of the Operational Research Society*, 59(6):743 – 755, 2008.

D. Leven and Z. Galil. NP completeness of finding the chromatic index of regular graphs. *Journal of Algorithms*, 4(1):35 – 44, 1983.

L. Lovász and M.D. Plummer. *Matching Theory*. North-Holland Mathematics Studies 121 / Annals of Discrete Mathematics 29. Elsevier Science Ltd, 1st edition, 1986.

R. Miyashiro, H. Iwasaki, and T. Matsui. Characterizing feasible pattern sets with a minimum number of breaks. In *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2002)*, volume 2740 of *LNCS*, pages 78 – 99. Springer, 2003.

D. Recalde, R. Torres, and P. Vaca. Scheduling the professional Ecuadorian football league by integer programming. *Computers & Operations Research*, 40(10):2478 – 2484, 2014.

J. Schönberger. Scheduling of sport league systems with inter-league constraints. In *Proceedings of the 5th International Conference on Mathematics in Sport*, pages 171 – 176, 2015.

T.A.M. Toffolo, J. Christiaens, F.C.R. Spieksma, and G. Vanden Berghe. The sport teams grouping problem. *Annals of Operations Research*, 275(1):223 – 243, 2019.